Available online at www.sciencedirect.com

SCIENCE DIRECT®

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Journal of Computational and Applied Mathematics 196 (2006) 58–69

www.elsevier.com/locate/cam

A Sinc-collocation method with boundary treatment for two-dimensional elliptic boundary value problems

Xionghua Wu*, Chen Li, Wenbin Kong

Department of Applied Mathematics, Tongji University, 200092 Shanghai, PR China

Received 18 March 2005

Abstract

It is difficult for the Sinc-collocation method to solve directly boundary value problems in two variables with the mixed nonhomogeneous boundary condition. In this paper, a developed Sinc-collocation method with boundary treatment (SCMBT) is introduced. It is easy to treat mixed nonhomogeneous boundary condition for our method. The error in the approximation of the solution is shown to converge at an exponential rate. And the numerical results show that compared with the exiting results, our method is of high accuracy, of good convergence with little computational efforts.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Sinc-collocation method; Singular perturbation; Boundary treatment; Convection–diffusion problems

1. Introduction

Sinc methods have been studied extensively and found to be a very effective technique, particularly for problems with singular solutions and those on unbounded domain. In addition, Sinc function seems to capture oscillating behaviors in space, hence, are useful to deal with problems characterized by this type of solution [11]. Lund and Bowers [2] and Stenger [5] provide overviews of the methods based on the Sinc function for solving ODE, PDE and integral equation. But it is difficult for the traditional Sinc method to solve two-dimensional elliptic boundary value problems with the mixed nonhomogeneous boundary condition [6]. In this paper, we present a Sinc-collocation method with boundaries treatment (SCMBT) based on the Sinc-collocation incorporated with the double exponential transformation technique. By our method, we can solve the PDE directly no matter what the boundary conditions are.

The double exponential formula (DE formula), which is a quadrature formula based on the double exponential transformation (DE transformation), was first proposed by Takshasi and Mori [9] in 1974. The DE formula has been widely used in the last three decades and is now recognized to be one of the most efficient quadrature formulas [3,8]. The use of the DE transformation technique in the Sinc method yields highly efficient numerical method for interpolation, quadrature, approximation of transformation, differential and partial differential equations [8,7,4].

It is known that the Sinc-collocation method with n collocation points converges at the rate of $\exp(-\kappa\sqrt{n})$ with some $\kappa > 0$ under certain conditions. From [8], we known that the Sinc-collocation incorporated with the DE transformation converge at the rate of $\exp(-\kappa' n/\log n)$ with some $\kappa' > 0$ under rather stringent conditions.

* Corresponding author. Tel.: +86 21 65980478; fax: +86 21 65982341.
E-mail addresses: wuxh@mail.tongji.edu.cn (X. Wu), chchlili@hotmail.com (C. Li).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.08.022

The mixed nonhomogeneous boundary conditions will bring difficulty for Sinc method to the numerical solutions of two-dimensional elliptic boundary value problems. As mentioned in [4], "…the derivatives of the Sinc functions can lead to a numerical overflow near the boundaries. One reason can be attributed to the tight grid spacing near the boundaries, where Sinc points become blustered in a geometric fashion."

By our method (SCMBT), we can overcome this difficulty easily. We can treat all kinds of boundary conditions directly and easily. We take the following steps. Let $\tilde{W}=(w(x_i, y_j))$ are the numerical solutions on the internal nodes of the two-dimensional elliptic boundary value problems. Except the $\tilde{W}=(w(x_i, y_j))$, the values on the boundary and the values of the normal derivative on the boundary are considered as the unknown quantities (in some cases, some of them are known). The values on the internal nodes $\tilde{W}$ and the values on the boundaries and the values of the normal derivatives on the boundaries constitute a new matrix W. With the help of the Coons' patch [1] and the calculation by the Sinc-collocation method, we can discretize the derivatives of w, such as $\partial w/\partial x$, $\partial w/\partial y$, $\partial^2 w/\partial x^2$, $\partial^2 w/\partial y^2$, $\partial^2 w/\partial x \partial y$, which can be denoted by expressions of W. Then the two-dimensional elliptic boundary value problems can be discretized into a matrix equation of W. With the block matrix technique, this matrix equation can be transformed into the block matrix equation of the values on the internal nodes $\tilde{W}$ and the unknown values on the boundaries and the unknown values on the normal derivatives of the boundaries. By the boundary conditions, the values on the boundaries and the values on the normal derivatives of the boundaries may be obtained, or the relation can be obtained. Substituting them into the block matrix equation, we can get the values on the internal nodes and the derivatives on the boundaries.

Thus, no matter what the boundary conditions are, we can solve the problems directly. Our method is based on Sinc-collocation and DE transformation, the numerical results show that compared with the existing results, our method is of high accuracy, of good convergence with little computational efforts.

The content of this paper is given in four sections. Section 1 is the Introduction; Section 2 introduces a new Sinc-collocation method (SCMBT); Section 3 applies our method to the numerical examples; and Section 4 is a short conclusion.

2. Numerical method

2.1. SCMBT formulas in one variable

From one of our papers [10], for a function $w(x)$ on the interval [0, 1], we can get the discretized formulas for $w_{xx} = f_1$, $w_x = f_2$.

Let $\varphi(x)$ be a one-to-one conformal map of interval [0, 1] onto the real line.

Denote $\mathbf{w} = [w(0), w_x(0), w(x_{-M}), \ldots, w(x_N), w_x(1), w(1)]^{\mathrm{T}}$, then $w_{xx} = f_1$ can be discretized as (details see [10])

$$\underline{B}\mathbf{w} = \overline{Q} F_1, \tag{2.1.1}$$

where

$$\underline{B} = [B_1, B_2, \tilde{B}, B_3, B_4], \quad F_1 = [f_1(x_{-M-1}), \ldots, f_1(x_{N+1})]^{\mathrm{T}}, \quad \overline{Q} = D\left(\frac{1}{\varphi'}\right)$$

$$B_1 = \overline{Q}\Phi''_{00} - \tilde{B}\tilde{\Phi}_{00}, \quad B_2 = \overline{Q}\Phi''_{01} - \tilde{B}\tilde{\Phi}_{01},$$

$$B_3 = \overline{Q}\Phi''_{11} - \tilde{B}\tilde{\Phi}_{11}, \quad B_4 = \overline{Q}\Phi''_{10} - \tilde{B}\tilde{\Phi}_{10}.$$

And here set $\varphi_{00}(x) = (2x+1)(1-x)^2$, $\varphi_{10}(x) = x^2(3-2x)$, $\varphi_{01}(x) = x(1-x)^2$, $\varphi_{11}(x) = x^2(x-1)$.

Let $\tilde{\Phi}_{ij} = [\varphi_{ij}(x_{-M}), \ldots, \varphi_{ij}(x_N)]^{\mathrm{T}}$, $\Phi''_{ij} = [\varphi''_{ij}(x_{-M-1}), \varphi''_{ij}(x_{-M}), \ldots, \varphi''_{ij}(x_N), \varphi''_{ij}(x_{N+1})]^{\mathrm{T}}$, where $i = 0, 1$, $j = 0, 1$.

$$\tilde{B} = \left[\frac{\tilde{I}^{(2)}}{h_x^2} + D\left(\frac{\varphi''}{(\varphi')^2}\right)\frac{\tilde{I}^{(1)}}{h_x} + D\left(\frac{1}{\varphi'}\left(\frac{1}{\varphi'}\right)''\right)\tilde{I}^{(0)}\right]\overline{D(\varphi')}.$$

Set $m = M + N + 1$. The matrix $\tilde{B}$ is a $(m+2) \times m$ matrix. The matrices $\tilde{I}^{(r)}$, $r = 0, 1, 2$ are $(m+2) \times m$ and the diagonal matrices $D(1/\varphi')$, $D(\varphi''/(\varphi')^2)$ and $D((1/\varphi')(1/\varphi')'')$ are $(m+2) \times (m+2)$ matrices. The diagonal matrix $\overline{D(\varphi')}$ is a $m \times m$ matrix. $\tilde{I}^{(r)} = (\delta_{jk}^{(r)})$, $k = -M, \ldots, N$ and $j = -M-1, -M, \ldots, N, N+1$.

Thus, $\underline{B}$ is a $(m+2) \times (m+4)$ matrix.

Similarly, $w_x = f_2$ can be discretized as (details see [4])

$$\underline{A}\mathbf{w} = \overline{Q}F_2, \tag{2.1.2}$$

where

$$\underline{A} = [A_1, A_2, \tilde{A}, A_3, A_4], \quad F_2 = [f_2(x_{-M-1}), \ldots, f_2(x_{N+1})]^{\mathrm{T}},$$

$$A_1 = \overline{Q}\Phi'_{00} - \tilde{A}\overline{\Phi}_{00}, \quad A_2 = \overline{Q}\Phi'_{01} - \tilde{A}\overline{\Phi}_{01},$$

$$A_3 = \overline{Q}\Phi'_{11} - \tilde{A}\overline{\Phi}_{11}, \quad A_4 = \overline{Q}\Phi'_{10} - \tilde{A}\overline{\Phi}_{10},$$

$$\tilde{A} = \left[-D\left(\frac{1}{\varphi'}\right)\frac{\tilde{I}^{(1)}}{h_x} + D\left(\frac{1}{\varphi'}\left(\left(\frac{1}{\varphi'}\right)'\right)\right)\tilde{I}^{(0)}\right]\overline{D((\varphi'))}.$$

2.2. The bicubically blended Coons' patch

In order to illustrate our method (SCMBT), Coons' patch is introduced.

Denote $\Omega = [0, h] \times [0, h]$. Let

$$\begin{cases} \varphi_{00} = (h-t)^2(h+2t)/h^3, \\ \varphi_{10} = t^2(3h-2t)/h^3, \\ \varphi_{01} = t(h-t)^2/h^3, \\ \varphi_{11} = t^2(h-t)/h^3. \end{cases}$$

For a function $u(x, y)$ satisfies:

$$\begin{cases} u(0, y) = f(0, y), \quad u(x, 0) = f(x, 0), \\ u(h, y) = f(h, y), \quad u(x, h) = f(x, h). \end{cases} \tag{2.2.1}$$

Applying the Hermite interpolation to $u(x, y)$ in x and y direction, respectively, we have:

$$\begin{cases} H_x u(x, y) = (\varphi_{00}(x), \varphi_{01}(x), \varphi_{11}(x), \varphi_{10}(x)) \begin{pmatrix} f(0, y) \\ f^{1,0}(0, y) \\ f^{1,0}(h, y) \\ f(h, y) \end{pmatrix}, \\ H_y u(x, y) = (f(x, 0), f^{0,1}(x, 0), f^{0,1}(x, h), f(x, h)) \begin{pmatrix} \varphi_{00}(y) \\ \varphi_{01}(y) \\ \varphi_{11}(y) \\ \varphi_{10}(y) \end{pmatrix}, \\ H_x H_y u(x, y) = (\varphi_{00}(x), \varphi_{01}(x), \varphi_{11}(x), \varphi_{10}(x)) M \begin{pmatrix} \varphi_{00}(y) \\ \varphi_{01}(y) \\ \varphi_{11}(y) \\ \varphi_{10}(y) \end{pmatrix}, \end{cases} \tag{2.2.2}$$

where $f^{1,0} = \partial f/\partial x$, $f^{0,1} = \partial f/\partial y$, $f^{1,1} = \partial^2 f/\partial x \partial y$ and

$$M = \begin{bmatrix} u(0, 0) & u_y(0, 0) & u_y(0, h) & u(0, h) \\ u_x(0, 0) & u_{xy}(0, 0) & u_{xy}(0, h) & u_x(0, h) \\ u_x(h, 0) & u_{xy}(h, 0) & u_{xy}(h, h) & u_x(h, h) \\ u(h, 0) & u_y(h, 0) & u_y(h, h) & u(h, h) \end{bmatrix}.$$

Then the Boolean sum defined by

$$P(x, y) = (H_x \oplus H_y)u(x, y) = H_x u(x, y) + H_y u(x, y) - H_x H_y u(x, y). \tag{2.2.3}$$

By direct calculation, we can get that

$$\begin{aligned}
[(H_x \oplus H_y)u(x, y)]|_{x=0} &= (1, 0, 0, 0) \begin{pmatrix} f(0, y) \\ f^{1,0}(0, y) \\ f^{1,0}(h, y) \\ f(h, y) \end{pmatrix} \\
&\quad + f(x, 0), f^{0,1}(x, 0), f^{0,1}(x, h), f(x, h)) \begin{pmatrix} \varphi_{00}(y) \\ \varphi_{01}(y) \\ \varphi_{11}(y) \\ \varphi_{10}(y) \end{pmatrix} \\
&\quad - (1, 0, 0, 0)M \begin{pmatrix} f(0, y) \\ f^{1,0}(0, y) \\ f^{1,0}(h, y) \\ f(h, y) \end{pmatrix} = f(0, y).
\end{aligned}$$

Similarly, we can know that $(H_x \oplus H_y)u(x, y)$ is the Coons patch, which satisfies:

$$\begin{cases} P_x(0, y) = f^{1,0}(0, y), & P_x(h, y) = f^{1,0}(h, y), \\ P_y(x, 0) = f^{0,1}(x, 0), & P_y(x, h) = f^{0,1}(x, h), \\ P_{xy}(0, y) = f^{1,1}(0, y), & P_{xy}(h, y) = f^{1,1}(h, y), \\ P_{xy}(x, 0) = f^{1,1}(x, 0), & P_{xy}(x, h) = f^{1,1}(x, h). \end{cases} \tag{2.2.4}$$

By [1], we can know that

$$|Ru(x, y)| = |u(x, y) - (H_x \oplus H_y)u(x, y)| \leqslant Ch^8.$$

2.3. A Sinc-collocation method with boundaries treatment (*SCMBT*)

In order to apply our method to the general cases of two-dimensional elliptic boundary value problems, first we will consider a particular case.

2.3.1. Sinc-collocation method for a particular case

In this part, we will give the discretized formulas for $u_x = f_1$, $u_y = f_2$, $u_{xy} = f_3$, $u_{xx} = f_4$, $u_{yy} = f_5$. when u satisfies the values and the normal derivatives on the boundaries of Ω are zeros.

Set the function $u(x, y)$ to be defined on $\Omega = [0, 1] \times [0, 1]$. Let $\xi = \varphi_x(x)$ and $\eta = \varphi_y(y)$ be the one-to-one conformal maps. Denote by $x = \psi_x(\xi)$ and $y = \psi_y(\eta)$ the inverses of the mappings φ_x and φ_y. Here φ_x, φ_y are double exponential transformations (details see [10]).

Define the change of variables:

$$v(\xi, \eta) = \varphi'_x \varphi'_y u = \varphi'_x(\psi_x(\xi))\varphi'_y(\psi_y(\eta))u(\psi_x(\xi), \psi_y(\eta)). \tag{2.3.1}$$

Thus

$$\lim_{\xi \to \pm\infty} v(\xi, \eta) = 0, \lim_{\eta \to \pm\infty} v(\xi, \eta) = 0.$$

Then we have

$$u_x=\frac{1}{\varphi_x'(x)\varphi_y'(y)}v_x-\frac{\varphi_x''(x)}{(\varphi_x'(x))^2}\frac{1}{\varphi_y'(y)}v=f_1, \tag{2.3.2}$$

$$u_y=\frac{1}{\varphi_x'(x)\varphi_y'(y)}v_y-\frac{\varphi_y''(y)}{(\varphi_y'(y))^2}\frac{1}{\varphi_x'(x)}v=f_2, \tag{2.3.3}$$

$$u_{xy}=\frac{1}{\varphi_x'(x)\varphi_y'(y)}v_{xy}-\frac{\varphi_x''(x)}{(\varphi_x'(x))^2}\frac{1}{\varphi_y'(y)}v_y-\frac{\varphi_y''(y)}{(\varphi_y'(y))^2}\frac{1}{\varphi_x'(x)}v_x+\frac{\varphi_x''(x)\varphi_y''(y)}{(\varphi_x'(x))^2(\varphi_y'(y))^2}v=f_3. \tag{2.3.4}$$

Writing

$$\frac{\partial v}{\partial x}=\frac{\partial v}{\partial \xi}\frac{\partial \xi}{\partial x}=\varphi_x'\frac{\partial v}{\partial \xi},\ \frac{\partial v}{\partial y}=\frac{\partial v}{\partial \eta}\frac{\partial \eta}{\partial y}=\varphi_y'\frac{\partial v}{\partial \eta},\ \frac{\partial^2 v}{\partial x\partial y}=\varphi_x'\varphi_y'\frac{\partial^2 v}{\partial \xi\partial \eta}. \tag{2.3.5}$$

Substitute Eq. (2.3.5) into Eq. (2.3.2)–(2.3.4), then we have

$$u_x=\frac{1}{\varphi_y'}v_\xi-\frac{\varphi_x''}{(\varphi_x')^2}\frac{1}{\varphi_y'}v=f_1, \tag{2.3.6}$$

$$u_y=\frac{1}{\varphi_x'}v_\eta-\frac{1}{\varphi_x'}\frac{\varphi_y''}{(\varphi_y')^2}v=f_2, \tag{2.3.7}$$

$$u_{xy}=v_{\xi\eta}-\frac{\varphi_y''}{(\varphi_y')^2}v_\xi-\frac{\varphi_x''}{(\varphi_x')^2}v_\eta+\frac{\varphi_x''\varphi_y''}{(\varphi_x')^2(\varphi_y')^2}v=f_3. \tag{2.3.8}$$

The assumed approximate solution takes the form

$$v_{m_x,m_y}(\xi,\eta)=\sum_{l=-M_y}^{N_y}\sum_{k=-M_x}^{N_x}v_{kl}S_{kl}(\xi,\eta), \tag{2.3.9}$$

where $v_{kl}=v(\xi_k,\eta_l)$, $m_x=M_x+N_x+1$, $m_y=M_y+N_y+1$. The basis function $\{S_{kl}(\xi,\eta)\}$ for $-M_x\leqslant k\leqslant N_x$, $-M_y\leqslant l\leqslant N_y$ are given as simple product basis function of the one-dimensional Sinc basis

$$S_{kl}(\xi,\eta)=[S(k,h_\xi)\circ\varphi_x][S(l,h_\eta)\circ\varphi_y].$$

With the properties of the Sinc function, applying Eq. (2.3.9) to Eq. (2.3.6) and evaluating the results at $\xi=ih_\xi$, $\eta=jh_\eta$, $-M_x-1\leqslant i\leqslant N_x+1$, $-M_y-1\leqslant j\leqslant N_y+1$, yields the identity

$$\begin{aligned}&\frac{1}{\varphi_x'(ih_\xi)(\varphi_y'(jh_\eta))^2}\frac{\partial v_{m_x,my}}{\partial x}(ih_\xi,jh_\eta)-\frac{\varphi_x''(ih_\xi)}{(\varphi_x'(ih_\xi))^3}\frac{1}{(\varphi_y'(jh_\eta))^2}v_{m_x,my}(ih_\xi,jh_\eta)\\&\quad=\frac{1}{\varphi_x'(ih_\xi)}f_1(ih_\xi,jh_\eta)\frac{1}{\varphi_y'(jh_\eta)},\end{aligned}$$

which upon letting $i=-M_x-1,\ldots,N_x+1$, $j=-M_y-1,\ldots,N_y+1$, has the more compact representation

$$\begin{aligned}&-\frac{1}{h_x}D\left(\frac{1}{\varphi_x'}\right)\tilde{I}_x^{(1)}\tilde{V}\tilde{I}_y^{(0)\mathrm{T}}D\left(\frac{1}{(\varphi_y')^2}\right)-D\left(\frac{\varphi_x''}{(\varphi_x')^3}\right)\tilde{I}_x^{(0)}\tilde{V}\tilde{I}_y^{(0)\mathrm{T}}D\left(\frac{1}{(\varphi_y')^2}\right)\\&\quad=D\left(\frac{1}{\varphi_x'}\right)F_1D\left(\frac{1}{\varphi_y'}\right),\end{aligned} \tag{2.3.10}$$

here $\tilde{V}=(v(\xi_i,\eta_j))_{m_x\times m_y}$, $i=-M_x,\ldots,N_x$, $j=-M_y,\ldots,N_y$, $F_1=(f_1(\xi_i,\eta_j))_{(m_x+2)\times(m_y+2)}$. Here, if a is a vector, then $D(a)$ denotes the diagonalization of a.

The matrices $\tilde{I}_x^{(r)}=(\delta_{ki}^{(r)})_{(m_x+2)\times m_x}$ are $(m_x+2)\times m_x$, and $\tilde{I}_y^{(r)}=(\delta_{lj}^{(r)})_{(m_y+2)\times m_y}$ are $(m_y+2)\times m_y$, $(r=0,1,2)$ (details see [10]). And the diagonal matrices $D(\varphi_x')$, $D(1/\varphi_x')$, $D(\varphi_x''/(\varphi')_x^3)$ are $(m_x+2)\times(m_x+2)$ matrices and the diagonal matrices $D(\varphi_y')$, $D(1/\varphi_y')$, $D(\varphi_y''/(\varphi')_y^3)$ are $(m_y+2)\times(m_y+2)$ matrices. The matrix $\overline{D(\varphi_x')}$ is a $m_x\times m_x$ matrix, and the matrix $\overline{D(\varphi_y')}$ is a $m_y\times m_y$matrix. $\tilde{I}_x^{(r)}=((\partial^r/\partial x^r)S_k(\xi_i))$, $k=-M,\ldots,N$ and $i=-M_x-1,-M_x,\ldots,N_x,N_x+1$, and $\tilde{I}_y^{(r)}=((\partial^r/\partial y^r)S_l(\eta_j))$, $l=-M_y,\ldots,N_y$ and $j=-M_y-1,-M_y,\ldots,N_y,N_y+1$. In fact $\tilde{I}_x^{(r)}$ is the remainder of $I_x^{(r)}=(\delta_{ki}^{(r)})_{(m_x+2)\times(m_x+2)}$ after removing the first and the last columns. And $\tilde{I}_y^{(r)}$ is the remainder of $I_y^{(r)}=(\delta_{lj}^{(r)})_{(m_y+2)\times(m_y+2)}$ after removing the first and the last columns.

Denote $\tilde{U}=(u(\xi_i,\eta_j))_{m_x\times m_y}$, $i=-M_x,\ldots,N_x$, $j=-M_y,\ldots,N_y$, then with Eq. (2.3.1) we have

$$\tilde{V}=\overline{D(\varphi_x')}\tilde{U}\overline{D(\varphi_y')}, \tag{2.3.11}$$

where the definitions of $\overline{D(\varphi_x')}$, $\overline{D(\varphi_y')}$ are same with the definitions above.

Denote $\overline{Q}_x=D(1/\varphi_x')$, $\overline{Q}_y=D(1/\varphi_y')$. Hence $\overline{Q}_x$ is a $(m_x+2)\times(m_x+2)$ matrix and $\overline{Q}_y$ is a $(m_y+2)\times(m_y+2)$ matrix.

Substituting Eq. (2.3.11) into Eq. (2.3.10), we can have

$$\begin{aligned}&-\frac{1}{h_x}D\left(\frac{1}{\varphi_x'}\right)\tilde{I}_x^{(1)}\overline{D(\varphi_x')}\tilde{U}\overline{D(\varphi_y')}\tilde{I}_y^{(0)\mathrm{T}}D\left(\frac{1}{(\varphi_y')^2}\right)\\&\quad-D\left(\frac{\varphi_x''}{(\varphi_x')^3}\right)\tilde{I}_x^{(0)}\overline{D(\varphi_x')}\tilde{U}\overline{D(\varphi_y')}\tilde{I}_y^{(0)\mathrm{T}}D\left(\frac{1}{(\varphi_y')^2}\right)=D\left(\frac{1}{\varphi_x'}\right)F_1D\left(\frac{1}{\varphi_y'}\right),\end{aligned}$$

i.e.

$$\left(-\frac{1}{h_x}D\left(\frac{1}{\varphi_x'}\right)\tilde{I}_x^{(1)}-D\left(\frac{\varphi_x''}{(\varphi_x')^3}\right)\tilde{I}_x^{(0)}\right)\overline{D(\varphi_x')}\tilde{U}\overline{D(\varphi_y')}\tilde{I}_y^{(0)\mathrm{T}}D\left(\frac{1}{(\varphi_y')^2}\right)=\overline{Q}_xF_1\overline{Q}_y.$$

Denote

$$\tilde{A}_x=\left(-\frac{1}{h_x}D\left(\frac{1}{\varphi_x'}\right)\tilde{I}_x^{(1)}-D\left(\frac{\varphi_x''}{(\varphi_x')^3}\right)\tilde{I}_x^{(0)}\right)\overline{D(\varphi_x')},\quad \tilde{Q}_y^{\mathrm{T}}=\overline{D(\varphi_y')}\tilde{I}_y^{(0)\mathrm{T}}D\left(\frac{1}{(\varphi_y')^2}\right), \tag{2.3.12}$$

thus, $\tilde{A}_x$ is a $(m_x+2)\times m_x$ matrix, and $\tilde{Q}_y^{\mathrm{T}}$ is $m_y\times(m_y+2)$.

So, $u_x=f_1$ can be discretized as

$$\tilde{A}_x\tilde{U}\tilde{Q}_y^{\mathrm{T}}=\overline{Q}_xF_1\overline{Q}_y. \tag{2.3.13}$$

Similarly, $u_y=f_2$, $u_{xy}=f_3$, $u_{xx}=f_4$ and $u_{yy}=f_5$ can be discretized. For example, $u_{xy}=f_3$ can be discretized as

$$\tilde{A}_x\tilde{U}\tilde{A}_y^{\mathrm{T}}=\overline{Q}_xF_3\overline{Q}_y, \tag{2.3.14}$$

where $\tilde{A}_y$ can be obtained after replacing $\varphi_x,\xi_i,h_x,\tilde{I}_x^{(r)},r=0,1$ in Eq. (2.3.12) by $\varphi_y,\eta_j,h_y,\tilde{I}_y^{(r)}$.

2.3.2. SCMBT formulas for general cases

In this part, we will discuss the discretized formula for $w_{xy}=f$, here $w(x,y)$ is not bound to be zero on the boundaries.

In order to express our method clearly, first we give some definitions.

Set $\tilde{\Phi}_{00}^x=(\varphi_{00}(x_i))_{m_x\times 1}$, $\tilde{\Phi}_{01}^x=(\varphi_{01}(x_i))_{m_x\times 1}$, $\tilde{\Phi}_{11}^x=(\varphi_{11}(x_i))_{m_x\times 1}$, $\tilde{\Phi}_{10}^x=(\varphi_{10}(x_i))_{m_x\times 1}$, $i=-M_x,\ldots,N_x$, $m_x=M_x+N_x+1$. $\tilde{\Phi}_{00}^y=(\varphi_{00}(y_j))_{m_y\times 1}$, $\tilde{\Phi}_{01}^y=(\varphi_{01}(y_j))_{m_y\times 1}$, $\tilde{\Phi}_{11}^y=(\varphi_{11}(y_j))_{m_y\times 1}$,$\tilde{\Phi}_{10}^y=(\varphi_{10}(y_j))_{m_y\times 1}$, $j=-M_y,\ldots,N_y, m_y=M_y+N_y+1$. $\varphi_{00},\varphi_{01},\varphi_{10},\varphi_{11}$ are the same with those in Section 2.2.

Set $\tilde{\Phi}_{x00}^x=(\varphi'_{00}(x_i))_{m_x\times 1}$, $\tilde{\Phi}_{x01}^x=(\varphi'_{01}(x_i))_{m_x\times 1}$, $\tilde{\Phi}_{x11}^x=(\varphi'_{11}(x_i))_{m_x\times 1}$, $\tilde{\Phi}_{x10}^x=(\varphi'_{10}(x_i))_{m_x\times 1}$, $i=-M_x,\ldots,N_x$, $m_x=M_x+N_x+1$. $\tilde{\Phi}_{y00}^y=(\varphi'_{00}(y_j))_{m_y\times 1}$, $\tilde{\Phi}_{y01}^y=(\varphi'_{01}(y_j))_{m_y\times 1}$,$\tilde{\Phi}_{y11}^y=(\varphi'_{11}(y_j))_{m_y\times 1}$, $\tilde{\Phi}_{y10}^y=(\varphi'_{10}(y_j))_{m_y\times 1}$, $j=-M_y,\ldots,N_y, m_y=M_y+N_y+1$.

Set $\Phi_{00}^x=(\varphi_{00}(x_i))_{(m_x+2)\times 1}$, $\Phi_{01}^x=(\varphi_{01}(x_i))_{(m_x+2)\times 1}$, $\Phi_{11}^x=(\varphi_{11}(x_i))_{(m_x+2)\times 1}$, $\Phi_{10}^x=(\varphi_{10}(x_i))_{(m_x+2)\times 1}$, $i=-M_x-1,\ldots,N_x+1, m_x=M_x+N_x+1$. $\Phi_{00}^y=(\varphi_{00}(y_j))_{(m_y+2)\times 1}$, $\Phi_{01}^y=(\varphi_{01}(y_j))_{(m_y+2)\times 1}$, $\Phi_{11}^y=(\varphi_{11}(y_j))_{(m_y+2)\times 1}$, $\Phi_{10}^y=(\varphi_{10}(y_j))_{(m_y+2)\times 1}$, $j=-M_y-1,\ldots,N_y+1, m_y=M_y+N_y+1$.

Set $\Phi_{x00}^x=(\varphi'_{00}(x_i))_{(m_x+2)\times 1}$, $\Phi_{x01}^x=(\varphi'_{01}(x_i))_{(m_x+2)\times 1}$, $\Phi_{x11}^x=(\varphi'_{11}(x_i))_{(m_x+2)\times 1}$, $\Phi_{x10}^x=(\varphi'_{10}(x_i))_{(m_x+2)\times 1}$, $i=-M_x-1,\ldots,N_x+1, m_x=M_x+N_x+1$. $\Phi_{y00}^y=(\varphi'_{00}(y_j))_{(m_y+2)\times 1}$, $\Phi_{y01}^y=(\varphi'_{01}(y_j))_{(m_y+2)\times 1}$, $\Phi_{y11}^y=(\varphi'_{11}(y_j))_{(m_y+2)\times 1}$, $\Phi_{y10}^y=(\varphi'_{10}(y_j))_{(m_y+2)\times 1}$, $j=-M_y-1,\ldots,N_y+1, m_y=M_y+N_y+1$.

Set $\tilde{W}=(w(x_l,y_k))_{m_x\times m_y}$, $w_{0j}=(w(0,y_k))$, $w_{x0j}=(w_x(0,y_k))$, $w_{x1j}=(w_x(1,y_k))$, $w_{1,j}=(w(1,y_k))$, $w_{i0}=(w(x_l,0))$, $w_{i1}=(w(x_l,1))$, $w_{xi0}=(w_x(x_l,0))$, $w_{xi1}=(w_x(x_l,1)), l=-M_x,\ldots N_x, k=-M_y,\ldots,N_y$. $w_{00}=w(0,0)$, $w_{01}=w(0,1)$, $w_{10}=w(1,0)$, $w_{11}=w(1,1)$. $w_{x00}=w_x(0,0)$, $w_{x01}=w_x(0,1)$, $w_{x10}=w_x(1,0)$, $w_{x11}=w_x(1,1)$. $w_{y00}=w_y(0,0)$, $w_{y01}=w_y(0,1)$, $w_{y10}=w_y(1,0)$, $w_{y11}=w_y(1,1)$. $w_{xy00}=w_{xy}(0,0)$, $w_{xy01}=w_{xy}(0,1)$, $w_{xy10}=w_{xy}(1,0)$, $w_{xy11}=w_{xy}(1,1)$. Here $\tilde{W}$ is a $m_x\times m_y$ matrix, $w_{0j}, w_{x0j}, w_{x1j}, w_{1,j}$ are row vectors , and $w_{i1}, w_{xi0}, w_{xi1}, w_{i0}$ are column vectors.

Set

$$W=\begin{bmatrix} w_{00} & w_{y00} & w_{0j} & w_{y01} & w_{01} \\ w_{x00} & w_{xy00} & w_{x0j} & w_{xy01} & w_{x01} \\ w_{i0} & w_{yi0} & \tilde{W} & w_{yi1} & w_{i1} \\ w_{x10} & w_{xy10} & w_{x1j} & w_{xy11} & w_{x11} \\ w_{10} & w_{y10} & w_{1j} & w_{y11} & w_{11} \end{bmatrix}. \tag{2.3.15}$$

For this function $w(x,y)$, set

$$u(x,y)=w(x,y)-(H_x w(x,y)+H_y w(x,y)-H_x H_y w(x,y)),$$

then

$$\frac{\partial^2}{\partial x\partial y}u=\frac{\partial^2}{\partial x\partial y}(w(x,y)-H_x w(x,y)-H_y w(x,y)+H_x H_y w(x,y)). \tag{2.3.16}$$

So we have

$$\frac{\partial^2}{\partial x\partial y}u=f-\frac{\partial^2}{\partial x\partial y}H_x w-\frac{\partial^2}{\partial x\partial y}H_y w(x,y)+\frac{\partial^2}{\partial x\partial y}H_x H_y w(x,y). \tag{2.3.17}$$

By Eq. (2.2.4) in Section 2.2, we know that $u=0$, $u_x=0$, $u_y=0$, $u_{xy}=0$ on the boundaries of Ω. So $(\partial^2/\partial x\partial y)u=g$ should be expressed as $\tilde{A}_x\tilde{U}\tilde{A}_y^{\mathrm{T}}=\overline{Q}_x G\overline{Q}_y$ by Eq. (2.3.14).

Thus, with the definitions of $H_x u$, $H_y u$ and $H_x H_y u$, Eq. (2.3.16) can be discretized as

$$\tilde{A}_x \tilde{U} \tilde{A}_y^{\mathrm{T}} = \tilde{A}_x \left(\tilde{W} - [\tilde{\Phi}_{00}^x, \tilde{\Phi}_{01}^x, \tilde{\Phi}_{11}^x, \tilde{\Phi}_{10}^x] \begin{bmatrix} w_{0j} \\ w_{x0j} \\ w_{x1j} \\ w_{1j} \end{bmatrix} - [w_{i0}, w_{yi0}, w_{yi1}, w_{i,1}] \begin{bmatrix} \tilde{\Phi}_{00}^{y\mathrm{T}} \\ \tilde{\Phi}_{01}^{y\mathrm{T}} \\ \tilde{\Phi}_{11}^{y\mathrm{T}} \\ \tilde{\Phi}_{10}^{y\mathrm{T}} \end{bmatrix} \right.$$
$$\left. - [\tilde{\Phi}_{00}^x, \tilde{\Phi}_{01}^x, \tilde{\Phi}_{11}^x, \tilde{\Phi}_{10}^x] M \begin{bmatrix} \tilde{\Phi}_{00}^{y\mathrm{T}} \\ \tilde{\Phi}_{01}^{y\mathrm{T}} \\ \tilde{\Phi}_{11}^{y\mathrm{T}} \\ \tilde{\Phi}_{10}^{y\mathrm{T}} \end{bmatrix} \right) \tilde{A}_y^{\mathrm{T}} = \tilde{A}_x \tilde{W} \tilde{A}_y^{\mathrm{T}} - \tilde{A}_x [\tilde{\Phi}_{00}^x, \tilde{\Phi}_{01}^x, \tilde{\Phi}_{11}^x, \tilde{\Phi}_{10}^x]$$
$$\times \begin{bmatrix} w_{0j} \\ w_{x0j} \\ w_{x1j} \\ w_{1j} \end{bmatrix} \tilde{A}_y^{\mathrm{T}} - \tilde{A}_x [w_{i0}, w_{yi0}, w_{yi1}, w_{i,1}] \begin{bmatrix} \tilde{\Phi}_{00}^{y\mathrm{T}} \\ \tilde{\Phi}_{01}^{y\mathrm{T}} \\ \tilde{\Phi}_{11}^{y\mathrm{T}} \\ \tilde{\Phi}_{10}^{y\mathrm{T}} \end{bmatrix} \tilde{A}_y^{\mathrm{T}} + \tilde{A}_x [\tilde{\Phi}_{00}^x, \tilde{\Phi}_{01}^x, \tilde{\Phi}_{11}^x, \tilde{\Phi}_{10}^x] M$$
$$\times \begin{bmatrix} \tilde{\Phi}_{00}^{y\mathrm{T}} \\ \tilde{\Phi}_{01}^{y\mathrm{T}} \\ \tilde{\Phi}_{11}^{y\mathrm{T}} \\ \tilde{\Phi}_{10}^{y\mathrm{T}} \end{bmatrix} \tilde{A}_y^{\mathrm{T}}, \tag{2.3.18}$$

where the definition of the matrix M is similar as in the definition in Section 2.2, but here take $h = 1$ as in Section 2.2.

Let A_x, A_y, B_x, B_y to be the weighting coefficient matrices, respectively, in x and y directions obtained by Eq. (2.1.1) and Eq. (2.1.2) of Section 2.1. With the formulas of Eq. (2.1.1), (2.1.2) and Eq. (2.3.13), (2.3.14), we can know that $\tilde{A}_x$, $\tilde{A}_y$ are just the remainders of A_x and A_y after removing the first, the second, the last second and the last columns, respectively. So, A_x, A_y are respectively split into the following block matrices:

$$A_x = [A_{x1}, A_{x2}, \tilde{A}_x, A_{x3}, A_{x4}], \quad A_y = [A_{y1}, A_{y2}, \tilde{A}_y, A_{y3}, A_{y4}], \tag{2.3.19}$$

here A_x is a $(m_x + 2) \times (m_x + 4)$ matrix, A_y is a $(m_y + 2) \times (m_y + 4)$ matrix, A_{x1}, A_{x2}, A_{x3}, A_{x4}, A_{y1}, A_{y2}, A_{y3}, A_{y4} are column vectors, $\tilde{A}_x$ is a $(m_x + 2) \times m_x$ matrix and $\tilde{A}_y$ is a $(m_y + 2) \times m_y$ matrix.

One the other hand, in Eq. (2.3.17) $H_x w$ can be considered as a univariate function of x and the derivatives in x direction of $H_x w$ can be gotten directly. And the derivatives in y direction of $H_x w$ can be gotten by the numerical method. Equivalently, for $(\partial^2/\partial x \partial y) H_x w$, we only approximate it in y direction by the formula of Section 2.1 and in x direction the derivatives can be gotten directly. $H_y w$ can be considered as a univariate function of y and the derivatives in y direction of $H_y w$ can be gotten directly. And the derivatives in x direction of $H_y w$ can be gotten by the numerical method. Equivalently, for $(\partial^2/\partial x \partial y) H_y w$, we only approximate it in x direction by the formula of Section 2.1 and in y direction the derivatives can be gotten directly. $H_x H_y w$ can be considered as a function of x and y. Thus, for $(\partial^2/\partial x \partial y) H_x w$, we can get it directly. So by (2.3.15) and Eq. (2.3.19), Eq. (2.3.17) can be discretized as follows:

$$\tilde{A}_x \tilde{U} \tilde{A}_y^{\mathrm{T}} = \overline{Q}_x F \overline{Q}_y - \overline{Q}_x [\Phi_{x00}^x, \Phi_{x01}^x, \Phi_{x11}^x, \Phi_{x10}^x] \begin{bmatrix} w_{00} & w_{y00} & w_{0j} & w_{y01} & w_{01} \\ w_{x00} & w_{xy00} & w_{x0j} & w_{xy01} & w_{x01} \\ w_{x10} & w_{xy10} & w_{x1j} & w_{xy11} & w_{x11} \\ w_{10} & w_{y10} & w_{1j} & w_{y11} & w_{11} \end{bmatrix}$$
$$\times \begin{bmatrix} A_{y1}^{\mathrm{T}} \\ A_{y2}^{\mathrm{T}} \\ \tilde{A}_y^{\mathrm{T}} \\ A_{y3}^{\mathrm{T}} \\ A_{y4}^{\mathrm{T}} \end{bmatrix} - [A_{x1}, A_{x2}, \tilde{A}_x, A_{x3}, A_{x4}] \begin{bmatrix} w_{00} & w_{y00} & w_{y01} & w_{01} \\ w_{x00} & w_{xy00} & w_{xy01} & w_{x01} \\ w_{i0} & w_{yi0} & w_{yi1} & w_{i1} \\ w_{x10} & w_{xy10} & uw_{xy11} & w_{x11} \\ w_{10} & w_{y10} & w_{y11} & w_{11} \end{bmatrix}$$
$$\times \begin{bmatrix} \Phi_{y00}^{y\mathrm{T}} \\ \Phi_{y01}^{y\mathrm{T}} \\ \Phi_{y11}^{y\mathrm{T}} \\ \Phi_{y10}^{y\mathrm{T}} \end{bmatrix} \overline{Q}_y + \overline{Q}_x [\Phi_{x00}^x, \Phi_{x01}^x, \Phi_{x11}^x, \Phi_{x10}^x] M \begin{bmatrix} \Phi_{y00}^{y\mathrm{T}} \\ \Phi_{y01}^{y\mathrm{T}} \\ \Phi_{y11}^{y\mathrm{T}} \\ \Phi_{y10}^{y\mathrm{T}} \end{bmatrix} \overline{Q}_y. \tag{2.3.20}$$

By Eq. (2.3.18), Eq. (2.3.20), we can get

$$\tilde{A}_x\tilde{W}\tilde{A}_y^{\mathrm{T}} - \tilde{A}_x[\tilde{\Phi}_{00}^x, \tilde{\Phi}_{01}^x, \tilde{\Phi}_{11}^x, \tilde{\Phi}_{10}^x]\begin{bmatrix} w_{0j} \\ w_{x0j} \\ w_{x1j} \\ w_{1j} \end{bmatrix}\tilde{A}_y^{\mathrm{T}}$$

$$- \tilde{A}_x[w_{i0}, w_{yi0}, w_{yi1}, w_{i,1}]\begin{bmatrix} \tilde{\Phi}_{00}^{y\mathrm{T}} \\ \tilde{\Phi}_{01}^{y\mathrm{T}} \\ \tilde{\Phi}_{11}^{y\mathrm{T}} \\ \tilde{\Phi}_{10}^{y\mathrm{T}} \end{bmatrix}\tilde{A}_y^{\mathrm{T}} + \tilde{A}_x[\tilde{\Phi}_{00}^x, \tilde{\Phi}_{01}^x, \tilde{\Phi}_{11}^x, \tilde{\Phi}_{10}^x]M\begin{bmatrix} \tilde{\Phi}_{00}^{y\mathrm{T}} \\ \tilde{\Phi}_{01}^{y\mathrm{T}} \\ \tilde{\Phi}_{11}^{y\mathrm{T}} \\ \tilde{\Phi}_{10}^{y\mathrm{T}} \end{bmatrix}\tilde{A}_y^{\mathrm{T}}$$

$$= \overline{Q}_x F\overline{Q}_y - \overline{Q}_x[\Phi_{x00}^x, \Phi_{x01}^x, \Phi_{x11}^x, \Phi_{x10}^x]\begin{bmatrix} w_{00} & w_{y00} & w_{0j} & w_{y01} & w_{01} \\ w_{x00} & w_{xy00} & w_{x0j} & w_{xy01} & w_{x01} \\ w_{x10} & w_{xy10} & w_{x1j} & w_{xy11} & w_{x11} \\ w_{10} & w_{y10} & w_{1j} & w_{y11} & w_{11} \end{bmatrix}\begin{bmatrix} A_{y1}^{\mathrm{T}} \\ A_{y2}^{\mathrm{T}} \\ \tilde{A}_y^{\mathrm{T}} \\ A_{y3}^{\mathrm{T}} \\ A_{y4}^{\mathrm{T}} \end{bmatrix}$$

$$- [A_{x1}, A_{x2}, \tilde{A}_x, A_{x3}, A_{x4}]\begin{bmatrix} w_{00} & w_{y00} & w_{y01} & w_{01} \\ w_{x00} & w_{xy00} & w_{xy01} & w_{x01} \\ w_{i0} & w_{yi0} & w_{yi1} & w_{i1} \\ w_{x10} & w_{xy10} & uw_{xy11} & w_{x11} \\ w_{10} & w_{y10} & w_{y11} & w_{11} \end{bmatrix}\begin{bmatrix} \Phi_{y00}^{y\mathrm{T}} \\ \Phi_{y01}^{y\mathrm{T}} \\ \Phi_{y11}^{y\mathrm{T}} \\ \Phi_{y10}^{y\mathrm{T}} \end{bmatrix}\overline{Q}_y$$

$$+ \overline{Q}_x[\Phi_{x00}^x, \Phi_{x01}^x, \Phi_{x11}^x, \Phi_{x10}^x]M\begin{bmatrix} \Phi_{y00}^{y\mathrm{T}} \\ \Phi_{y01}^{y\mathrm{T}} \\ \Phi_{y11}^{y\mathrm{T}} \\ \Phi_{y10}^{y\mathrm{T}} \end{bmatrix}\overline{Q}_y.$$

By some direct calculations, we have $\underline{A}_x W\underline{A}_y^{\mathrm{T}} = \overline{Q}_x F\overline{Q}_y$, where

$$\underline{A}_x = [\overline{Q}_x\Phi_{x00}^x - \tilde{A}_x\tilde{\Phi}_{00}^x,\ \overline{Q}_x\Phi_{x01}^x - \tilde{A}_x\tilde{\Phi}_{01}^x,\ \tilde{A}_x,\ \overline{Q}_x\Phi_{x11}^x - \tilde{A}_x\tilde{\Phi}_{11}^x,\ \overline{Q}_x\Phi_{x10}^x - \tilde{A}_x\tilde{\Phi}_{10}^x],$$

$$\underline{A}_y = [\overline{Q}_y\Phi_{y00}^y - \tilde{A}_y\tilde{\Phi}_{00}^y,\ \overline{Q}_y\Phi_{y01}^y - \tilde{A}_y\tilde{\Phi}_{01}^y,\ \tilde{A}_y,\ \overline{Q}_y\Phi_{y11}^y - \tilde{A}_y\tilde{\Phi}_{11}^y,\ \overline{Q}_y\Phi_{y10}^y - \tilde{A}_y\tilde{\Phi}_{10}^y],$$

they are exactly the same with A_x, A_y obtained by Eq. (2.1.1) and Eq. (2.1.2) in *x* direction and *y* direction, respectively.

Similarly, $w_y = f_2$, $w_x = f_3$, $w_{yy} = f_4$ and $w_{xx} = f_5$ can be discretized as, respectively:

$$\underline{Q}_x W\underline{A}_y^{\mathrm{T}} = \overline{Q}_x F_2\overline{Q}_y, \underline{A}_x W\underline{Q}_y = \overline{Q}_x F_3\overline{Q}_y,$$

$$\underline{Q}_x W\underline{B}_y^{\mathrm{T}} = \overline{Q}_x F_4\overline{Q}_y, \underline{B}_x W\underline{Q}_y = \overline{Q}_x F_5\overline{Q}_y,$$

where $\underline{Q}_x$, $\underline{Q}_y$, $\underline{B}_x$, $\underline{B}_y$ are the same with the matrices obtained by applying our formulas for one variable in *x* direction and *y* direction respectively (details see [10]).

2.4. Application of SCMBT

The following problem is considered:

$$a(x, y)w_{xx} + b(x, y)w_{xy} + c(x, y)w_{yy} + d(x, y)w_x + e(x, y)w_y + f(x, y)w(x, y) = g(x, y). \tag{2.4.1}$$

Discretize Eq. (2.4.1) by Eq. (2.3.8), we have:

$$C_1 \circ (\underline{B}_x W\underline{Q}_y) + C_2 \circ (\underline{A}_x W\underline{A}_y^{\mathrm{T}}) + C_3 \circ (\underline{Q}_x W\underline{B}_y^{\mathrm{T}}) + C_4 \circ (\underline{A}_x W\underline{Q}_y)$$
$$+ C_5 \circ (\underline{Q}_x W\underline{A}_y^{\mathrm{T}}) + C_6 \circ (\underline{Q}_x W\underline{Q}_y) = \overline{Q}G\overline{Q}, \tag{2.4.2}$$

where '$\circ$' is the Hadamard product of the matrices. $C_1=(a(x_i, y_j))$, $C_2=(b(x_i, y_j))$, $C_3=(c(x_i, y_j))$, $C_4=(d(x_i, y_j))$, $C_5=(e(x_i, y_j))$, $C_6=(f(x_i, y_j))$, $G=(g(x_i, y_j))$, $i=-M_x-1, \ldots, N_x+1$, $j=-M_y-1, \ldots, N_y+1$. And W is the same as Eq. (2.3.15).

Here, the boundary conditions can be easily treated. For example, if boundary condition is

$$w_x(0, y)+w(0, y)=1, \quad 0 \leqslant y \leqslant 1$$

it means that in Eq. (2.3.15) $w_{0j}+w_{x0j}=1$, $w_{00}+w_{x00}=1$ and $w_{01}+w_{x01}=1$.

With the block matrix technique, we can separate the known block elements of Eq. (2.3.15) from the unknown block elements. Then with the boundary conditions, we can get the numerical solutions. So by our method we can discretize the equation and the boundary conditions no matter what the boundary conditions are.

3. Numerical examples and analysis

In this part, two examples are treated in order to illustrate the accuracy of our method. And we use the following parameters in the double exponential transformations of both the x and y directions, $\beta=\pi/2$, $\gamma=2$, $d=\pi/4$, which lead to $h_x=\log(m_x\pi)/2m_x$, $h_x=\log(m_y\pi)/2m_y$.

In this part, the maximum of the absolute errors are calculated, namely Maxerror.

Example 1. Consider the problem:

$$\begin{cases} e^x w_{xx}+e^{xy} w_{xy}+e^y w_{yy}=f, \\ w(0, y)=e^y, \quad \left.\left(\dfrac{\partial w}{\partial x}+w\right)\right|_{x=1}=2e^{1+y}, \\ w(x, 0)=e^x, \quad \left.\left(\dfrac{\partial w}{\partial y}\right)\right|_{y=1}=e^{1+x}. \end{cases} \tag{3.1}$$

Here assume the analytical solution of Eq. (3.1) is $w(x, y)=e^{x+y}$. Thus, we can get the function $f(x, y)$.

Discretize Eq. (3.1) by our method directly. By the boundary conditions, we have $w_{00}=1$, $w_{0j}=(e^{y_j})$, $w_{01}=e$, $w_{i0}=(e^{x_i})$, $w_{10}=e$, $w_{y00}=1$, $w_{x00}=1$, $w_{x10}=e$, $w_{y01}=e$, $w_{yi1}=(e^{x_i+1})$, $w_{y11}=e^2$, $w_{xy01}=e$, $w_{xy11}=e^2$, $w_{x11}+w_{11}=2e^2$, $w_{xy10}+w_{y10}=2e$, $w_{x1j}+w_{1j}=2e^{1+y_j}$. Substituting these into the discretized matrix equation, we can get the numerical solutions.

Table 1 shows the numerical solutions. Here take $M_x=M_y$, $N_x=N_y$, $m_x=m_y$. Table 2 shows the numerical results for the normal derivatives on the boundaries. The numerical results show that our method is of high accuracy, of good convergence with little computational efforts. With our method, we can get the numerical results for derivatives on the boundaries easily and effectively.

Example 2. In this example, our method is applied to the two-dimensional singular perturbation problem as follows:

$$\begin{cases} -\varepsilon w_{xx}-\varepsilon w_{yy}+w_w+w_y=f, & (x, y) \in \Omega=(0, 1) \times (0, 1), \\ w=0, & (x, y) \in \partial\Omega. \end{cases} \tag{3.2}$$

Here we take the analytic solution $w=xy(1-e^{(x-1)/\varepsilon})(1-e^{(y-1)/\varepsilon})$. By the boundary conditions, we known that W of Eq. (2.3.15) in the first row, the first column, the last row and the last column are zero. Here take $M_x=M_y$, $N_x=N_y$, $m_x=m_y$. Thus, we can get the numerical results which are shown in Table 3. The results in [12] is shown in Table 4. In [12], the finite element combined with the domain decomposition was used and the domain was divided into four sub-domains. M in Table 4 for [12] represents the number of points in x direction in every sub-domain. In [12], the number of points in the y direction is the same as the number of points in the x direction. In Table 3 M_x is the number of points in the x direction of the whole domain. Compared with Table 4, Table 3 shows that by our method without domain decomposition we can get much better numerical results. When $\varepsilon=0.01, 0.001$, the numerical results are very good. If $\varepsilon=0.0001$, we need more nodes to get satisfactory results. In this case, the domain decomposition method should be introduced. It is convenient to

Table 1
The absolute errors for Example 1

M_x	4	6	8	10	12
Maxerror	1.2785e−004	4.1127e−005	1.6450e−005	7.2493e−006	3.4191e−006

Table 2
The absolute errors of the normal derivatives on the boundaries for Example 1

M_x	4	6	8	10	12
Maxerror $at\ x=0$	6.3527e−004	2.3142e−004	1.0240e−004	4.8983e−005	2.4745e−005
Maxerror $at\ x=1$	1.2805e−004	4.1140e−005	1.6452e−005	7.2495e−006	3.4192e−006
Maxerror $at\ y=0$	5.2768e−004	1.9644e−004	8.8359e−005	4.2790e−005	2.1806e−005

Table 3
The absolute errors for Example 2

M_x	MaxError ($\varepsilon=0.01$)	MaxError ($\varepsilon=0.001$)	Max Error ($\varepsilon=0.0001$)
11	3.7833e−004	0.1386	15.7531
15	1.1695e−004	0.0154	1.2329
23	3.7766e−007	7.2022e−004	0.0202
31	3.3206e−008	4.7864e−005	0.0105

Table 4
The numerical results with domain decomposition in [12]

M	MaxError ($\varepsilon=0.01$)	MaxError ($\varepsilon=0.001$)	Max Error($\varepsilon=0.0001$)
12	0.007428	0.010876	0.011487
16	0.005522	0.006288	0.006742
24	0.003023	0.003068	0.003188
32	0.002019	0.002039	0.002042
48	0.001120	0.001131	0.001132

Table 5
The absolute errors for Example 2 with domain decomposition

	$M_x=8$	$M_x=10$	$M_x=12$	$M_x=16$	$M_x=20$
$\text{MaxError}_{\varepsilon=0.0001}$	0.0365	4.3035e−004	7.5939e−005	1.9671e−005	4.0147e−006

combine our method with the domain decomposition method, because the values on the inner boundaries and the values of the normal derivatives on the inner boundaries are contained as unknowns in Eq. (2.3.15). We decomposite the domain into four sub-domains: $[0, 1-\delta]\times[0, 1-\delta]$, $[0, 1-\delta]\times[1-\delta, 1]$, $[1-\delta, 1]\times[0, 1-\delta]$, $[1-\delta, 1]\times[1-\delta, 1]$, δ=0.0012. The results are very satisfactory. Table 5 shows the numerical results with the domain decomposition. In Table 5 M_x is the number of the points in the x direction of every sub-domain, and take $M_x = M_y$, $N_x = N_y$, $m_x = m_y$ in every sub-domain.

4. Conclusions

In this paper, a new Sinc-collocation method with the boundary conditions treatment (SCMBT) is presented for two-dimensional elliptic boundary value problems. By our method, we can discretize the equations directly no matter what the boundary conditions are. For the boundary conditions we do not have to do more work, which may decrease the accuracy of the method. Our method is shown to be of good convergence, simple in principle, easy to program, easy to treat the boundary conditions. SCMBT can be easily used to the Sinc-collocation patching domain decomposition methods. The convergence analysis of SCMBT can be found in the paper [10].

References

[1] R.E. Barnhill, W. Boehm, Surfaces in CAGD'84, Elsevier, Amsterdam, 1985.
[2] J. Lund, K. Bowers, Sinc Methods for Quadrature and Differential Equations, SIAM, Philadelphia, PA, 1992.
[3] M. Mori, M. Sugihara, The double exponential transformation in numerical analysis, in: W. Gautschi, F. Marcellan, L. Reichel (Eds.), Numerical Analysis in the 20th Century, vol. V: Qudarature and Orthogonal Polynomials, J. Comput. Appl. Math. 127 (2001) 287–296.
[4] S. Narasimhan, J. Majdalani, F. Stenger, A first step in applying the Sinc collocation method to the nonlinear Navier–Stokes Equations, Numer. Heat Transfer, Part B 41 (2002) 447–462.
[5] F. Stenger, Numerical Methods Based on Sinc and Analytic Functions, Springer, New York, 1993.
[6] F. Stenger, Summary of Sinc numerical methods, in: L. Wuytack, J. Wimp (Eds.), Numerical Analysis in the 20th Century, vol. I, Approximation Theory, J. Comput. Appl. Math. 121 (2000) 379–420.
[7] M. Sugihara, Double exponential transformation in the Sinc-collocation method for two-point boundary value problems, J. Comput. Appl. Math. 149 (2002) 239–250.
[8] M. Sugihara, T. Matsuo, Recent developments of the Sinc numerical methods, J. Comput. Appl. Math. 164–165 (2004) 673–689.
[9] H. Takahasi, M. Mori, Double exponential formulas for numerical integration, Publ. Res. Inst. Math. Sci. 9 (1974) 721–741.
[10] X. Wu, W. Kong, C. Li, Sinc collocation method with boundary treatment for two-point boundary value problem, J. Comput. Appl. Math., to appear, doi: 10.1016/j.cam.2005.09.003.
[11] Y. Wu, X. Wu, Linearized and rational approximation method for solving nonlinear Burgers' equation, Internat. J. Numer. Methods Fluids 45 (2004) 509–525.
[12] Z. Zhang, Finite element superconvergence on Shishkin mesh for 2-D convection–diffusion problems, Math. Comput. 72 (2003) 1147–1177.